

# ON COMPUTATIONAL PATHS AND THE FUNDAMENTAL GROUPOID OF A TYPE

Arthur F. Ramos<sup>\*</sup>  
afr@cin.ufpe.br

Ruy J. G. B. de Queiroz<sup>\*</sup>  
ruy@cin.ufpe.br

Anjolina G. de Oliveira<sup>\*</sup>  
ago@cin.ufpe.br

The main objective of this work is to study mathematical properties of computational paths. Originally proposed by de Queiroz & Gabbay (1994) as ‘sequences of rewrites’, computational paths can be seen as the grounds on which the propositional equality between two computational objects stand. Using computational paths and categorical semantics, we take any type  $A$  of type theory and construct a groupoid for this type. We call this groupoid the fundamental groupoid of a type  $A$ , since it is similar to the one obtained using the homotopical interpretation of the identity type. The main difference is that instead of being just a semantical interpretation, computational paths are entities of the syntax of type theory. We also expand our results, us-

---

<sup>\*</sup> Centro de Informática, Universidade Federal de Pernambuco, Recife-PE, Brazil

ing computational paths to construct fundamental groupoids of higher levels.

**Keywords:** Computational paths, fundamental groupoid, equality theory, term rewriting systems, type theory, category theory, higher categorical structures.

## 1 INTRODUCTION

There seems to be little doubt that the identity type is one of the most intriguing concepts of Martin-Löf's Type Theory. This claim is supported by recent groundbreaking discoveries. In 2005, Vladimir Voevodsky [1] discovered the Univalent Models, resulting in a new area of research known as Homotopy Type Theory [2]. This theory is based on the fact that a term of some identity type, for example  $p : \text{Id}_A(a, b)$ , has a clear homotopical interpretation. The interpretation is that the witness  $p$  can be seen as a homotopical path between the points  $a$  and  $b$  within a topological space  $A$ . This simple interpretation has made clear the connection between Type Theory and Homotopy Theory, generating groundbreaking results, as one can see in [3, 2]. It is important to emphasize that one important fact of the homotopic interpretation is that the homotopic paths exist only in the semantic sense. In other words, there is no formal entity in the language of type theory that represents these paths. They are not present in the syntax of Type Theory.

Given a space  $A$ , we can think of a structure  $\Pi(A)$  formed by points of the space  $A$  and homotopical paths between these points. Using straightforward equations, one can easily prove for  $\Pi(A)$  that the groupoid equations hold up to homotopy. We call this *weak* structure (it is weak since the equalities do not hold 'on the nose', but only up to homotopy) the fundamental groupoid of the space  $A$ . Since each type  $A$ , using the homotopical interpretation, can be semantically interpreted as being a space, we can consider  $\Pi(A)$  as being the fundamental groupoid of the type  $A$ .

As we already mentioned, the paths of the homotopical interpretation do not have a counterpart in the syntax of type theory, but exist only as semantical interpretations. With that in mind, adding the concept of paths to the syntax of type theory was the focus of some of our recent works [4, 5, 6]. In these works, we argued that it is possible to formalize the concept

of path as an formal entity of the equality theory. This formal entity, known as computational paths, establishes the equality between two terms of the same type. In fact, we argued that the identity type is just the type of this new entity. That way, we added the concept of path to the syntax of type theory, instead of considering it only as a semantic interpretation.

Our main objective in this work is to construct a fundamental groupoid for a type  $A$  using the concept of computational path, yet having the same semantics as in the homotopical interpretation. We will also go a little further, showing that it is possible to construct higher groupoid structures, like the weak 2-groupoid of a type  $A$  usually written as  $\Pi_2(A)$ .

## 2 COMPUTATIONAL PATHS

As previously mentioned, computational paths will be the main focus here. For that reason, a good understanding of the meaning of this entity is essential. The concept of computational path is inspired on the equality theory of  $\lambda$ -calculus, known as  $\lambda\beta\eta$  – equality [7]. This theory establishes when two  $\lambda$ -terms are equal. It establishes a set of axioms and rules of inference that one can use. The idea is that a term  $A$  is  $\lambda\beta\eta$  – equal to a term  $B$  when one can reach  $B$  from  $A$  after the application of a set of axioms and rules of inference. The application of these axioms and rules of inference generates a path. As a very simple example, since  $(\lambda x.x)z \triangleright_\beta z$ , then we say that  $(\lambda x.x)z =_{\beta\eta} z$  because of the path  $\beta((\lambda x.x)z, z)$ . A more complex and interesting example will be showed after we formally define a computational path. Since  $\lambda\beta\eta$ -equality establishes equalities between terms of the  $\lambda$ -calculus, we need to find an equivalent theory for Type Theory, which establishes equality between terms of the same type. To do that, we just need to translate to Type Theory the axioms and rules of inference of the  $\lambda\beta\eta$ -equality. For example, for the product type  $\Pi$ , we obtain the following axioms (one can check the original axioms of  $\lambda\beta\eta$  – equality in [7]):

$$\begin{array}{ll}
 (\beta) \quad \frac{[x : A] \quad N : A \quad M : B}{(\lambda x.M)N = M[N/x] : B[N/x]} & (\xi) \quad \frac{[x : A] \quad M = M' : B}{\lambda x.M = \lambda x.M' : (\Pi x : A)B}
 \end{array}$$

$$\begin{array}{ll}
(\rho) \quad \frac{M : A}{M = M : A} & (\mu) \quad \frac{M = M' : A \quad N : (\Pi x : A)B}{NM = NM' : B[M/x]} \\
(\sigma) \quad \frac{M = N : A}{N = M : A} & (\nu) \quad \frac{N : A \quad M = M' : (\Pi x : A)B}{MN = M'N : B[N/x]} \\
(\tau) \quad \frac{M = N : A \quad N = P : A}{M = P : A} & \\
(\eta) \quad \frac{M : (\Pi x : A)B}{(\lambda x. Mx) = M : (\Pi x : A)B} \quad (x \notin \text{FV}(M)) & 
\end{array}$$

Aside from these axioms, we can also apply a change of bounded variables. To simplify the notation, if not previously mentioned, consider that each computational path that appears in this work is considered modulo change of bounded variables.

**Definition 2.1.** *Let  $a$  and  $b$  be elements of a type  $A$ . Then, a computational path  $s$  from  $a$  to  $b$  is a composition of applications of the inference rules of the equality theory of type theory. We denote that by  $a =_s b$ .*

The composition is given by applications of the transitive rule. We can better understand this fact by analyzing an example. consider the construction of the path between  $(\lambda y. yx)(\lambda w. zw)$  and  $zx$ . Starting from  $(\lambda y. yx)(\lambda w. zw)$ , we have that  $(\lambda y. yx)(\lambda w. zw) \triangleright_\eta (\lambda y. yx)z$ . Then, we have the path  $\eta((\lambda y. yx)(\lambda w. zw), (\lambda y. yx)z)$ . From  $(\lambda y. yx)z$ , we have that  $(\lambda y. yx)z \triangleright_\beta zx$ . Therefore, we have a path  $\beta((\lambda y. yx)z, zx)$ . We now need to compose these paths. To do this, we can apply the transitivity to obtain the desired path between  $(\lambda y. yx)(\lambda w. zw)$  and  $zx$ . Hence, the equality is established by the path  $\tau(\eta((\lambda y. yx)(\lambda w. zw), (\lambda y. yx)z), \beta((\lambda y. yx)z, zx))$ . So, in the end, a computational path is a term built out of the composition of (definitional) equality identifiers. A notion of canonical (path) identifier will be defined, and a set of reduction rules for (path) terms will be given explicitly.

In our recent works, we argue that computational paths are in fact the formal inhabitants of the identity type, since they can be seen as a ‘derivation-term’ of a proof of propositional equality. Since a path  $a =_s b : A$  establishes the equality between  $a$  and  $b$ , then we can think of  $s(a, b)$  as being an object of  $\text{Id}_A(a, b)$ . One can check [4, 5, 6] for more details.

### 3 REDUCTIONS BETWEEN COMPUTATIONAL PATHS

In the previous section, we showed that computational paths establishes the equality between two terms of the same type. In this section, our objective is to investigate if two seemingly different computational paths could be considered equals, i.e., if there is the possibility of establishing a path between paths. Consider the simple case of a reflexive path  $\rho$  that establishes that  $a =_\rho a$ . If we apply the axiom of symmetry  $\sigma$ , we obtain the path  $a =_{\sigma(\rho)} a$ . Since we applied the axiom  $\sigma$  to the trivial case of reflexivity, we obtained a path that is just a redundant form of the reflexivity. In that case, the paths  $\rho$  and  $\sigma(\rho)$  should be considered equals, i.e., it should be possible to reduce  $\sigma(\rho)$  to just  $\rho$ . The same thing happens if, given any path  $a =_s b$ , we apply the symmetry two times in succession, obtaining the path  $a =_{\sigma(\sigma(s))} b$ . Inverting a path two times is equivalent as just using the original path, i.e.,  $\sigma(\sigma(s))$  should be reduced to  $s$ .

As one could notice, the aforementioned redundancies have been originated from combination of applications of the axioms of the equality theory for type theory (as instance, the redundancy of our first example has been originated by the combination of applications of reflexivity and symmetry). In the previous section, using the example of the product type  $\Pi$ , we showed that we have a total of 8 equality axioms. Since we can combine these axioms in different situations, one should expect that the number of possible redundancies might be high. Fortunately, all these redundancies and reductions have already been mapped by [8] and further developed (and given identifiers) by [4]. That way, [8] created a system that maps all these reductions, called  $\text{LND}_{\text{EQ}} - \text{TRS}$ , with a total of 39 reduction rules. To construct the fundamental groupoid, 7 rules will be essential. They are as follows:

- Reductions involving  $\sigma$  and  $\rho$ :  

$$\sigma(\rho) \triangleright_{\text{sr}} \rho \quad \sigma(\sigma(r)) \triangleright_{\text{ss}} r$$
- Reductions involving  $\tau$   

$$\tau(r, \sigma(r)) \triangleright_{\text{tr}} \rho \quad \tau(\sigma(r), r) \triangleright_{\text{tsr}} \rho$$

$$\tau(r, \rho) \triangleright_{\text{trr}} r \quad \tau(\rho, r) \triangleright_{\text{tlr}} r$$
- Reductions involving  $\tau$  and  $\tau$

$$\tau(\tau(t, r), s) \triangleright_{tt} \tau(t, \tau(r, s))$$

As one can see, an identifier is given to each reduction rule. For example, our initial examples are resolved by applications of the rule *sr* and the rule *ss* respectively.

It is important to note that the system  $\text{LND}_{\text{EQ}} - \text{TRS}$ , is terminating and confluent, as proved in [8, 9, 10, 11].

We call each reduction rule as a rewrite rule and usually abbreviated to *rw-rule*. Associated to an *rw-rule*, we have the following definitions:

**Definition 3.1.** *Let  $s$  and  $t$  be computational paths. We say that  $s \triangleright_{1rw} t$  (read as:  $s$  *rw-contracts to*  $t$ ) iff we can obtain  $t$  from  $s$  by an application of only one *rw-rule*. If  $s$  can be reduced to  $t$  by finite number of *rw-contracts*, then we say that  $s \triangleright_{rw} t$  (read as  $s$  *rw-reduces to*  $t$ ).*

**Definition 3.2.** *Let  $s$  and  $t$  be computational paths. We say that  $s =_{rw} t$  (read as:  $s$  is *rw-equal to*  $t$ ) iff  $t$  can be obtained from  $s$  by a finite (perhaps empty) series of *rw-contracts* and reversed *rw-contracts*. In other words,  $s =_{rw} t$  iff there exists a sequence  $R_0, \dots, R_n$ , with  $n \geq 0$ , such that*

$$\begin{aligned} (\forall i \leq n-1) (R_i \triangleright_{1rw} R_{i+1} \text{ or } R_{i+1} \triangleright_{1rw} R_i) \\ R_0 \equiv s, \quad R_n \equiv t \end{aligned}$$

By definition, the *rw-equality* is a class of equivalence, since it was defined as the symmetric, reflexive and transitive closure of *rw-reduction*. Since an *rw-equality* is given by a sequence of *rw-reductions*, we call this sequence an *rw-sequence*.

### 3.1 Reduction between *rw-equalities*

We showed that it is possible to think of reductions between paths and that these reductions were originated from redundancies caused by the combination of the equality axioms. Since *rw-equalities* are transitive, symmetric and reflexive, it is possible to think of reductions between two *rw-sequences*. For example, if  $\theta$  is an *rw-sequence*, we can think of  $\sigma(\sigma(\theta))$ , which should be equivalent to  $\theta$ . In fact, knowing that the *ss* rule resolves this kind of redundancy for computational paths, we can think of a *ss*<sub>2</sub> rule that resolves this redundancy for *rw-sequences*. In fact, since our rules that we showed previously only involve transitivity, reflexivity and symmetry, and that *rw-equality* has these properties, then we get an *rw-equality* version of each rule, only adding the label 2 to make clear that we

are working with  $\text{rw}$ -sequences and  $\text{rw}$ -equality. In fact, analogously to  $\text{rw}$ -equality, we can think of a  $\text{rw}_2$ -equality, which has the same definition, with the difference that we are working with reductions between  $\text{rw}$ -sequences. For that reason,  $\text{rw}_2$ -equality is also an equivalence class.

Since  $\text{rw}_2$ -equality forms a new system of rules, it is possible to think of rules that are specific to  $\text{rw}_2$ -equality. In fact, there is a specific rule that will play an essential role in the construction of our higher fundamental groupoid  $\Pi_2(A)$ . Consider the example of the path  $\tau(s, t)$ , such that  $s \triangleright_{1\text{rw}} s'$  and  $t \triangleright_{1\text{rw}} t'$ . There are two possible  $\text{rw}$ -sequences:  $\tau(s, t) \triangleright_{1\text{rw}} \tau(s', t) \triangleright_{1\text{rw}} \tau(s', t')$  and  $\tau(s, t) \triangleright_{1\text{rw}} \tau(s, t') \triangleright_{1\text{rw}} \tau(s', t')$ . As one can see, the only difference between these two  $\text{rw}$ -sequences are the choice of which term to develop first. Since  $s$  and  $t$  followed the same reductions (i.e.,  $t$  reduced to  $t'$  in both cases and  $s$  to  $s'$ ), then these two  $\text{rw}$ -sequences are two ways of expressing the same result. For that reason, there should be a rule that establishes the  $\text{rw}_2$ -equality of these  $\text{rw}$ -sequences. We call this rule the independence of choice inside the transitivity, and it is expressed by  $\text{cd}_2$ . Then, we have that  $(\tau(s, t) \triangleright_{1\text{rw}} \tau(s', t) \triangleright_{1\text{rw}} \tau(s', t')) =_{\text{rw}_2\text{cd}} (\tau(s, t) \triangleright_{1\text{rw}} \tau(s, t') \triangleright_{1\text{rw}} \tau(s', t'))$ . As one could see, this rule only makes sense in the context of  $\text{rw}$ -sequences and, for this reason, it is specific to  $\text{rw}_2$ -equality and  $\text{rw}_2$ -reduction.

## 4 THE FUNDAMENTAL GROUPOID OF A TYPE: $\Pi(A)$

In this section, our objective is to show that the computational paths are capable of constructing the fundamental groupoid of a type  $A$ . The first thing that we should be aware is the idea of equality holding in a weak sense. In the homotopical interpretation, the idea of weak is related to a structure which the equalities only hold up to homotopy. In this work, we will use the term weak to characterize a structure which equalities only hold up to  $\text{rw}$ -equality or  $\text{rw}_2$ -equality. If the equalities hold 'on the nose', instead of being weak, we say that the structure is strict.

First, let's recall some basic concepts of category theory [12]:

**Definition 4.1.** Let  $f : A \rightarrow B$  be an arrow of any category.  $f$  is called an isomorphism if there exists a  $g : B \rightarrow A$  such that  $g \circ f = 1_A$  and  $f \circ g = 1_B$ .  $g$  is called the inverse of  $f$  and can be written as  $f^{-1}$ .

**Definition 4.2.** *A groupoid is a category in which every arrow is an isomorphism.*

We can now think of a structure  $\Pi(A)$  in which the objects are the objects  $a : A$  and the morphisms are the computational paths between these objects:

**Proposition 4.1.**  *$\Pi(A)$  is a weak groupoid.*

*Proof.* First, we need to prove that  $\Pi(A)$  is a weak category. To do that, we need to define composition of morphisms and the identity arrow. We already know that composition of paths  $s \circ t$  is given by application of the transitivity, i.e.,  $s \circ t = \tau(t, s)$ . The identity morphism of an object  $a$  is given by the reflexive path  $a =_{\rho_a} a$ . We need now to check the associative and identity laws. The associativity equation holds weakly, we just need to use the  $tt$  rule:

$$\tau(\tau(s, r), t) =_{rw_{tt}} \tau(s, \tau(r, t)).$$

Using rules  $tlr$  and  $trr$ , we show that the identity laws also hold weakly:

$$\begin{aligned} s \circ 1_a &= s \circ \rho_a = \tau(\rho_a, s) =_{rw_{tlr}} s \\ 1_b \circ s &= \rho_b \circ s = \tau(s, \rho_b) =_{rw_{trr}} s \end{aligned}$$

With these conditions satisfied, we conclude that  $\Pi(A)$  is indeed a weak category. To conclude our proof, we need to show that it also is a weak groupoid. To do that, we need to show that every computational path  $s$  has a inverse computational path  $s'$ . Finding  $s'$  is easy, just put  $s' = \sigma(s)$ . To show that the equalities of the isomorphism hold weakly, we use rules  $tr$  and  $tsr$ :

$$\begin{aligned} s \circ s' &= s \circ \sigma(s) = \tau(\sigma(s), s) =_{rw_{tsr}} \rho_b \\ s' \circ s &= \sigma(s) \circ s = \tau(s, \sigma(s)) =_{rw_{tr}} \rho_a \end{aligned}$$

With that, we conclude that  $\Pi(A)$  is a weak groupoid. □

Since the groupoid  $\Pi(A)$  is constructed from a type  $A$  and its objects and equalities between these objects (given by computational paths), we define  $\Pi(A)$  as the fundamental groupoid of type  $A$ .

As proved by [13], the groupoid model of a type proves that the uniqueness of identity proofs is not derivable in the syntax of type theory. The construction of the fundamental groupoid



of a type based on computational paths makes clear that the addition of computational paths to the syntax of type theory will still refute the uniqueness of identity proofs. In this sense, one can interpret this result as the fact that there may be two computational paths  $s$  and  $t$  between objects  $a : A$  and  $b : A$  such that  $s$  and  $t$  are not  $\text{rw}$ -equal.

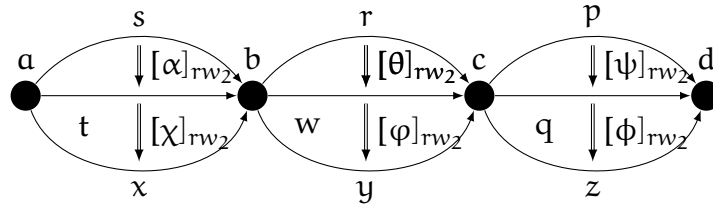
#### 4.1 The fundamental 2-groupoid $\Pi_2(A)$ of a type

Using the homotopical interpretation, one can construct a weak 2-groupoid of a space  $A$ . This 2-groupoid, called the fundamental 2-groupoid of the space  $A$ , can be constructed by taking  $\Pi(A)$  and adding a groupoid structure to each pair of objects of  $\Pi(A)$ . In this sense, for each pair of objects  $a, b \in \Pi(A)$ , we have a groupoid  $\Pi(a, b)$ , in which the objects are homotopical paths between  $a$  and  $b$  and morphisms are classes of homotopies between these paths. It is a well established fact that  $\Pi(A)$  together with the structures  $\Pi(a, b)$  forms a weak 2-groupoid, known as  $\Pi_2(A)$  [14].

The objective of this subsection is to show that it is possible to construct, using computational paths, the fundamental 2-groupoid  $\Pi_2(A)$  of a type  $A$ . To construct  $\Pi_2(A)$ , we can think of the structure  $\Pi(A)$  with the addition of substructures  $\Pi(A)(a, b)$  between each pair of objects of  $\Pi(A)$ . In  $\Pi(A)(a, b)$ , objects are computational paths between  $a : A$  and  $b : A$  and morphisms are  $\text{rw}$ -equalities modulo  $\text{rw}_2$ -equality.  $\Pi(a, b)$  is a strict groupoid. The proof of this fact is analogous to **proposition 4.1**. The difference is that instead of using  $\text{rw}$ -rules, this proof uses the equivalent  $\text{rw}_2$ -rule. Another difference is that the equations hold strictly, since the morphisms are modulo  $\text{rw}_2$ -equality. Now we can prove the following proposition:

**Proposition 4.2.** *Given any type  $A$ ,  $\Pi_2(A)$  is a weak 2-groupoid.*

*Proof.* First of all, we can represent  $\Pi_2(A)$  graphically as follows:



We already know that  $\Pi(A)$  is a weak groupoid and that  $\Pi(A)(a, b)$  is a strict groupoid. To finish the proof that  $\Pi_2(A)$  is a weak 2-groupoid, we need to show that  $\Pi_2(A)$  is a bicategory.

As one can check in [14], the proof that a structure is a bicategory follows a considerable number of steps. To achieve this proof, one of the first steps is to define how a 2-morphism (as we already know, in  $\Pi_2(A)$  2-morphisms are  $\text{rw}$ -equalities modulo  $\text{rw}_2$ -equality) can be composed horizontally. Given an  $\text{rw}$ -sequence  $[\alpha]_{\text{rw}_2} : [s = \alpha_1, \alpha_2, \dots, \alpha_n = t]_{\text{rw}_2}$  and an  $\text{rw}$ -sequence  $[\theta]_{\text{rw}_2} : [r = \theta_1, \theta_2, \dots, \theta_m = w]_{\text{rw}_2}$ , we define the horizontal composition  $[\alpha]_{\text{rw}_2} \circ_h [\theta]_{\text{rw}_2}$  as the following  $\text{rw}$ -sequence:

$$[\alpha]_{\text{rw}_2} \circ_h [\theta]_{\text{rw}_2} = [\tau(s, r) = \tau(\alpha_1, \theta_1), \tau(\alpha_2, \theta_1), \dots, \tau(\alpha_n = t, \theta_1), \tau(\alpha_n, \theta_2), \dots, \tau(\alpha_n, \theta_m) = \tau(t, w)]_{\text{rw}_2}$$

With the horizontal composition well defined, we need to check if it is associative and respect the identity laws. For the associativity, we need to show that there is a natural isomorphism  $\text{assoc}$  between  $(([\psi]_{\text{rw}_2} \circ_h [\theta]_{\text{rw}_2}) \circ_h [\alpha]_{\text{rw}_2})$  and  $([\psi]_{\text{rw}_2} \circ_h ([\theta]_{\text{rw}_2} \circ_h [\alpha]_{\text{rw}_2}))$ . To prove that, we will use the well established fact that a natural transformation is a natural isomorphism iff every component is an isomorphism [12]. That way, we only need to prove that there is an isomorphism between every component of  $(([\psi]_{\text{rw}_2} \circ_h [\theta]_{\text{rw}_2}) \circ_h [\alpha]_{\text{rw}_2})$  and  $([\psi]_{\text{rw}_2} \circ_h ([\theta]_{\text{rw}_2} \circ_h [\alpha]_{\text{rw}_2}))$ . In fact, since we are working in a groupoid structure, we just need to check that there is a morphism between each component (since every morphism is an isomorphism in a groupoid). To do this, we can get a generic component  $\tau(\alpha_x, \tau(\theta_y, \psi_z))$  of  $(([\psi]_{\text{rw}_2} \circ_h [\theta]_{\text{rw}_2}) \circ_h [\alpha]_{\text{rw}_2})$  ( $x, y, z$  being suitable naturals that respect the order of the horizontal composition) and show that it has a morphism to the equivalent component  $\tau(\tau(\alpha_x, \theta_y), \psi_z)$  of  $([\psi]_{\text{rw}_2} \circ_h ([\theta]_{\text{rw}_2} \circ_h [\alpha]_{\text{rw}_2}))$ . This morphism is established by  $\sigma(\text{tt})$  as follows:

$$\tau(\alpha_x, \tau(\theta_y, \psi_z)) =_{\text{rw}_{\sigma(\text{tt})}} \tau(\tau(\alpha_x, \theta_y), \psi_z)$$

To prove the identity law, we use a similar idea. We need to check the natural isomorphism  $r_s^*$  between  $[\alpha]_{\text{rw}_2} \circ_h [\rho_{\rho_a}]_{\text{rw}_2}$  and  $[\alpha]_{\text{rw}_2}$ . To do that, we prove find a morphism between components  $\tau(\rho_{\rho_a}, \alpha_y)$  and  $\alpha_y$ :

$$\tau(\rho_{\rho_a}, \alpha_y) =_{\text{rw}_{\text{ttr}}} \alpha_y.$$

We now need to check the second identity law, i.e., the natural isomorphism  $l_s^*$  between  $([\rho_{\rho_b}]_{\text{rw}_2} \circ_h [\alpha]_{\text{rw}_2})$  and  $[\alpha]_{\text{rw}_2}$ . Taking components  $\tau(\alpha_y, \rho_{\rho_a})$  and  $\alpha_y$ :

$$\tau(\alpha_y, \rho_{\rho_a}) =_{\text{rw}_{\text{tr}}} \alpha_y.$$

Also associated to the horizontal composition, we need now to check the interchange law, i.e., we need to check:

$$([\varphi]_{\text{rw}_2} \circ [\theta]_{\text{rw}_2}) \circ_h ([\chi]_{\text{rw}_2} \circ [\alpha]_{\text{rw}_2}) = ([\varphi]_{\text{rw}_2} \circ_h [\chi]_{\text{rw}_2}) \circ ([\theta]_{\text{rw}_2} \circ_h [\alpha]_{\text{rw}_2})$$

From  $(([\varphi]_{\text{rw}_2} \circ [\theta]_{\text{rw}_2}) \circ_h ([\chi]_{\text{rw}_2} \circ [\alpha]_{\text{rw}_2}))$ , we have:

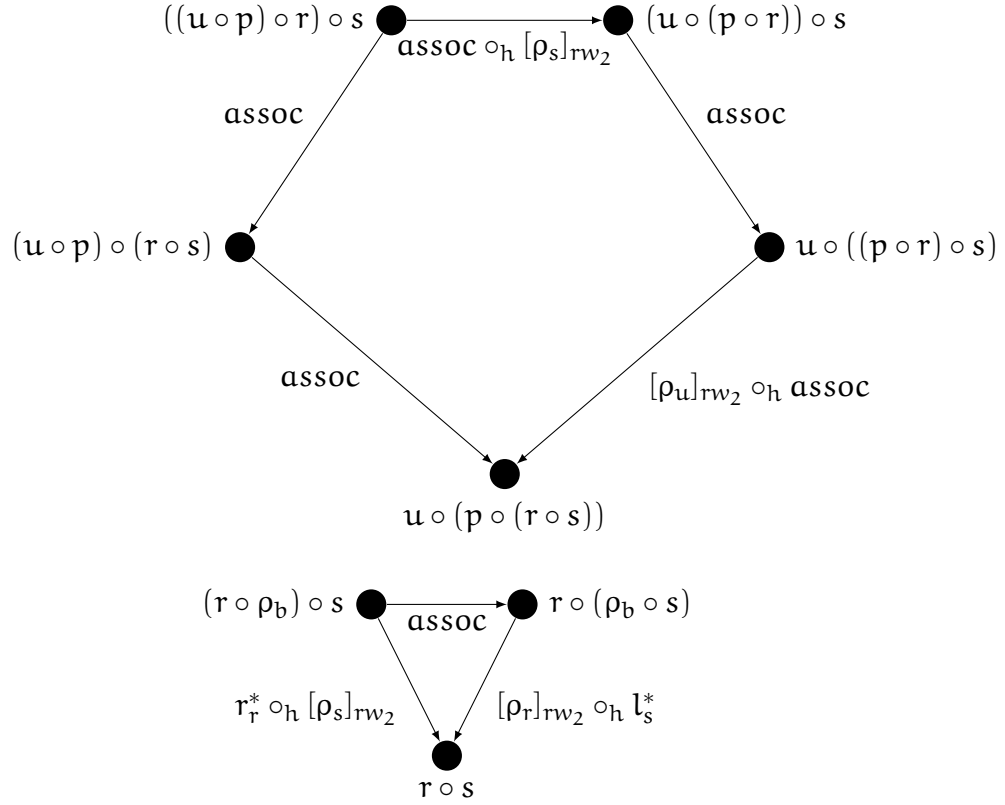
$$\begin{aligned} & (([\varphi]_{\text{rw}_2} \circ [\theta]_{\text{rw}_2}) \circ_h ([\chi]_{\text{rw}_2} \circ [\alpha]_{\text{rw}_2})) = \\ & [\tau(\theta, \varphi)]_{\text{rw}_2} \circ_h [\tau(\alpha, \chi)]_{\text{rw}_2} = \\ & [\theta_1, \dots, \theta_n = \varphi_1, \dots, \varphi_{n'}]_{\text{rw}_2} \circ_h [(\alpha_1, \dots, \alpha_m = \chi_1, \dots, \chi_{m'})]_{\text{rw}_2} = \\ & [\tau(\alpha_1, \theta_1), \dots, \tau(\alpha_m = \chi_1, \theta_1), \dots, \tau(\chi_n, \theta_1), \dots, \tau(\chi_n, \theta_{m'} = \\ & \varphi_1), \dots, \tau(\chi_n, \varphi_{n'})]_{\text{rw}_2} \end{aligned}$$

From  $(([\varphi]_{\text{rw}_2} \circ_h [\chi]_{\text{rw}_2}) \circ ([\theta]_{\text{rw}_2} \circ_h [\alpha]_{\text{rw}_2}))$ :

$$\begin{aligned} & (([\varphi]_{\text{rw}_2} \circ_h [\chi]_{\text{rw}_2}) \circ ([\theta]_{\text{rw}_2} \circ_h [\alpha]_{\text{rw}_2}))(\mathbf{r} \circ \mathbf{s}) = \\ & ([\tau(\chi_1, \varphi_1), \dots, \tau(\chi_n, \varphi_1), \dots, \tau(\chi_n, \varphi_{n'})]_{\text{rw}_2} \circ \\ & [\tau(\alpha_1, \theta_1), \dots, \tau(\alpha_m, \theta_1), \dots, \tau(\alpha_m, \theta_{m'})]_{\text{rw}_2}) = \\ & [\tau(\alpha_1, \theta_1), \dots, \tau(\alpha_m, \theta_1), \dots, \tau(\alpha_m, \theta_{m'}), \dots, \tau(\chi_1, \varphi_1), \dots, \\ & \tau(\chi_n, \varphi_1), \dots, \tau(\chi_n, \varphi_{n'})]_{\text{rw}_2} \end{aligned}$$

As one can see, the sole difference between the two results is the order that the internal  $\text{rw}$ -sequences have been developed in each transitivity. This is a perfect case to use the independence of choice inside the transitivity, i.e., the  $\text{rw}_2$ -rule known as  $\text{cd}_2$ . By one application of  $\text{cd}_2$ , we conclude that  $[\tau(\alpha_1, \theta_1), \dots, \tau(\alpha_m = \chi_1, \theta_1), \dots, \tau(\chi_n, \theta_1), \dots, \tau(\chi_n, \theta_{m'} = \varphi_1), \dots, \tau(\chi_n, \varphi_{n'})]_{\text{rw}_2} = [\tau(\alpha_1, \theta_1), \dots, \tau(\alpha_m, \theta_1), \dots, \tau(\alpha_m, \theta_{m'}), \dots, \tau(\chi_1, \varphi_1), \dots, \tau(\chi_n, \varphi_1), \dots, \tau(\chi_n, \varphi_{n'})]_{\text{rw}_2}$ .

To end our proof, since we are working with a weak structure, we need to check the coherence laws for a bicategory. The coherence laws are given by Mac Lane's pentagon and triangle [14]:



We need to show that the diagrams above commute. The equations are straightforward. Starting with the pentagon, we can start from  $((u \circ p) \circ r) \circ s = \tau(s, \tau(r, \tau(p, u)))$  and go to the right of the diagram:

$$\begin{aligned}
 & (\text{assoc} \circ_h [\rho_s]_{rw_2})(\tau(s, \tau(r, \tau(p, u)))) = \\
 & \quad \tau(s, \text{assoc}(\tau(r, \tau(p, u)))) = \\
 & \quad \tau(s, \tau(\tau(r, p), u)) \\
 & \text{assoc}(\tau(s, \tau(\tau(r, p), u))) = \tau(\tau(s, \tau(r, p)), u) \\
 & ([\rho_u]_{rw_2} \circ_h \text{assoc})(\tau(\tau(s, \tau(r, p)), u)) = \\
 & \quad \tau(\text{assoc}(\tau(s, \tau(r, p))), u) = \\
 & \quad \tau(\tau(\tau(s, r), p), u) = u \circ (p \circ (r \circ s))
 \end{aligned}$$

Starting from the same  $((u \circ p) \circ r) \circ s = \tau(s, \tau(r, \tau(p, u)))$  and going bottom left:

$$\begin{aligned}
 & \text{assoc}(\tau(s, \tau(r, \tau(p, u)))) = \tau(\tau(s, r), \tau(p, u)) \\
 & \text{assoc}(\tau(\tau(s, r), \tau(p, u))) = \tau(\tau(\tau(s, r), p), u) = u \circ (p \circ (r \circ s))
 \end{aligned}$$

We conclude that the pentagon commutes. For the triangle, we can start with  $(r \circ \rho_b) \circ s = \tau(s, \tau(\rho_b, r))$  and go to the right:

$$\begin{aligned}
\text{assoc}(\tau(s, \tau(\rho_b, r))) &= \tau(\tau(s, \rho_b), r) \\
([\rho_r]_{\text{rw}_2} \circ_h l_s^*) \tau(\tau(s, \rho_b), r) &= \tau(l_s^*(\tau(s, \rho_b)), r) = \\
\tau(s, r) &= r \circ s
\end{aligned}$$

Now starting with the same  $((r \circ \rho_b) \circ s) = \tau(s, \tau(\rho_b, r))$  and going to to bottom right:

$$\begin{aligned}
(r_r^* \circ_h [\rho_s]_{\text{rw}_2}) \tau(s, \tau(\rho_b, r)) &= \tau(s, r_r^*(\tau(\rho_b, r))) = \\
\tau(s, r) &= r \circ s
\end{aligned}$$

The triangle also commutes. The coherence laws hold. With that, we finish the proof that  $\Pi_2(A)$  is a weak 2-groupoid.  $\square$

With this proof, we conclude that, using computational paths, it is possible to construct the fundamental 2-groupoid of a type  $A$ . In fact, using the same reasoning that led us to come up with  $\text{rw}_2$ -equality, one could think of  $\text{rw}_3$ -equality,  $\text{rw}_4$ -equality, etc. That way, it would be possible to think of the possibility of even higher fundamental groupoids. If we continue this process to the infinite, we would think of a possible weak infinite groupoid  $\Pi_\infty(A)$ , also called weak  $\omega$ -groupoid. As a matter of fact, it has already been proved that it is possible to obtain such structure, as shown in [15, 16]. Our objective, in a future work, is to obtain similar results using the concept of computational paths. The same way that we used to construct the fundamental 2-groupoid  $\Pi_2(A)$ , we could use computational paths and infinite levels of  $\text{rw}$ -equalities to construct  $\Pi_\infty(A)$ . The problem is that  $\Pi_\infty(A)$  is a complex structure and to prove that it is a weak  $\omega$ -groupoid is not an easy task. For that reason, obtaining this construct will be the main focus of some of our future works.

## 5 CONCLUSION

The main objective of this work was to construct a groupoid model for a type using the concept of computational paths. Based on the idea that it is possible to introduce an entity known as computational paths to the syntax of type theory, we used this newly added entity to construct a groupoid model for a type  $A$ . We have showed that this approach is fundamentally different from the groupoid constructed using homotopical paths, since computational paths are de facto elements of the syntax of type theory, instead of being only semantical interpretations.

To achieve our results, we have showed that it is possible to think of reductions between two different computational paths. Based on these reductions, we have constructed a structure in which objects are elements of a type  $A$  and morphisms are computational paths. We have showed that this structure is, in fact, a weak groupoid, and called it the fundamental groupoid of a type  $A$ . We have gone further, defining a higher structure and proving that this structure is also a groupoid. That way, using computational paths, we have obtained the fundamental 2-groupoid of the type  $A$ . A rather natural way forward is to show that it is indeed possible to obtain even higher structures, with the main goal of using computational paths to obtaining a possible fundamental  $\infty$ -groupoid.

## REFERENCES

- [1] V. Voevodsky, “Univalent foundations and set theory.” Univalent Foundations and Set Theory, Lecture at IAS, Princeton, New Jersey, Mar 2014.
- [2] S. Awodey, “Type theory and homotopy,” in *Epistemology versus Ontology* (P. Dybjer, S. Lindström, E. Palmgren, and G. Sundholm, eds.), vol. 27 of *Logic, Epistemology, and the Unity of Science*, pp. 183–201, Springer Netherlands, 2012.
- [3] T. Univalent Foundations Program, *Homotopy Type Theory: Univalent Foundations of Mathematics*. Institute for Advanced Study: <http://homotopytypetheory.org/book>, 2013.
- [4] R. J. G. B. de Queiroz and A. G. de Oliveira, “Propositional equality, identity types, and direct computational paths,” 2011. <http://arxiv.org/abs/1107.1901>, 2011 (latest version 2013).
- [5] A. F. Ramos, R. J. G. B. de Queiroz, and A. G. de Oliveira, “On the identity type as the type of computational paths,” 2015. <http://arxiv.org/abs/1504.04759>, 2015. Accepted for presentation in LSFA 2015 - 10th Workshop on Logical and Semantic Frameworks, with Applications, Natal, RN, 31-Ago a 01-Set, 2015.
- [6] R. J. G. B. de Queiroz and A. G. de Oliveira, “Natural deduction for equality: The missing entity,” in *Advances in*

- Natural Deduction* (L. Pereira, E. Haeusler, and V. de Paiva, eds.), pp. 63–91, Springer, 2014.
- [7] J. R. Hindley and J. P. Seldin, *Lambda-calculus and combinators: an introduction*. Cambridge University Press, 2008.
  - [8] A. G. de Oliveira, “Proof transformations for labelled natural deduction via term rewriting,” 1995. Master’s thesis, Depto. de Informática, Universidade Federal de Pernambuco, Recife, Brazil, April 1995.
  - [9] A. G. de Oliveira and R. J. G. B. de Queiroz, “Term rewriting with labelled deductive systems,” in *Proceedings of Brazilian Symposium on Artificial Intelligence (SBIA’94)*, pp. 59–72, 1994.
  - [10] A. G. de Oliveira and R. J. G. B. de Queiroz, “A normalization procedure for the equational fragment of labelled natural deduction,” *Logic Journal of IGPL*, vol. 3, no. (2–3), pp. 243–290, 1999.
  - [11] R. J. G. B. de Queiroz, A. G. de Oliveira, and D. M. Gabbay, *The Functional Interpretation of Logical Deduction*. World Scientific, 2011.
  - [12] S. Awodey, *Category theory*. Oxford University Press, 2nd ed., 2010.
  - [13] M. Hofmann and T. Streicher, “The groupoid model refutes uniqueness of identity proofs,” in *Logic in Computer Science, 1994. LICS’94. Proceedings., Symposium on*, pp. 208–212, IEEE, 1994.
  - [14] T. Leinster, *Higher Operads, Higher Categories*. London Mathematical Society Lecture Note Series (Book 298), Cambridge University Press, 2004. Also <http://arxiv.org/abs/math/0305049>, May, 2003.
  - [15] P. L. Lumsdaine, “Weak  $\omega$ -categories from intensional type theory,” in *Typed lambda calculi and applications*, vol. 5608 of *LNCS*, pp. 172–187, Springer, 2009.
  - [16] B. van den Berg and R. Garner, “Types are weak  $\omega$ -groupoids,” *Proceedings of the London Mathematical Society*, vol. 102, no. 2, pp. 370–394, 2011.